

FINITE AUTOMATA WITH RESTRICTED TWO-WAY MOTION

DAVID DAMANIK

ABSTRACT. We consider finite two-way automata and measure the use of two-way motion by counting the number of left moves in accepting computations. Restriction of the automata according to this measure allows us to study in detail the use of two-way motion for the acceptance of regular languages in terms of state complexity. The two-way spectrum of a given regular language is introduced. This quantity reflects the change of size of minimal accepting devices if the use of two-way motion is increased incrementally. We give examples for spectra, prove uniform upper and lower bounds and study their sharpness. We also have state complexity results for two-way automata with uniformly bounded use of two-way motion.

1. INTRODUCTION

The basic acceptance model for regular languages is the deterministic finite one-way automaton ($1DFA$). There are two natural ways of extending this model, allowing nondeterminism and two-way motion of the input head, respectively. Both of them do not allow the acceptance of non-regular languages. However, the extensions ($1NFA$ and $2DFA$) can produce exponential savings in the number of states required to recognize a regular language.

State complexity issues have a long history going back at least to the pioneering work [12] of Meyer and Fischer. After equivalence of the several models of finite automata was established by Rabin-Scott [14], Shepherdson [16] and Vardi [18], there was a considerable interest in results which demonstrate that a certain state complexity blow-up is indeed necessary in some cases. We mention the works of Sakoda-Sipser [15], Sipser [17], and Micali [13].

On the other hand, in some cases not the whole power of the extension is needed, for some languages the minimal $1DFA$ is as small as the minimal $1NFA$ (resp., $2DFA$). Thus, for a given regular language, one can ask how much nondeterminism (resp., two-way motion) is required to describe it adequately. Of course, one has to introduce well-motivated measures for nondeterminism (resp., two-way motion) first.

In their 1990 paper [7], Goldstine, Kintala and Wotschke handled the case of nondeterminism. The measure they used reflects the maximal (taken over the words in the language) number of branches in the minimal (regarding the number of branches) accepting computation. Starting with a $1DFA$ and allowing more and more nondeterminism according to this measure they studied the decrease of state complexity in the nondeterminism spectrum. Concerning two-way motion, we are going to follow similar lines.

By counting the number of left moves in the accepting computations we measure the two-way complexity of a given automaton. We get the class $2DFA(k)$ by restricting $2DFA$ according to this measure. An automaton from the class $2DFA(k)$

uses at most k left moves in an accepting computation. The two-way spectrum of a regular language L is then given by a monotonically decreasing infinite sequence where the entries are the sizes of minimal $2DFA(k)$ -descriptions of L . The rate of decrease reflects the change of size of a $2DFA$ describing L when the two-way complexity is increased incrementally.

The paper is organized as follows. Section 2 introduces notation and recalls some classical results. Section 3 introduces our two-way complexity measure, the class $2DFA(k)$, and the spectrum $\sigma(L)$ of a regular language L , gives some examples of spectra with certain interesting properties, proves uniform upper and lower bounds for spectra and studies their sharpness. Section 4 compares the class $2DFA(k)$ with the classes $1DFA$, $1NFA$ and $2DFA$ under the viewpoint of possible state complexity savings (or, equivalently, blow-ups).

Due to the analogy mentioned above, some ideas and techniques are inspired by [7]. Nevertheless, for the sake of self-containedness, full proofs of our theorems are provided.

2. FINITE TWO-WAY AUTOMATA

2.1. Definitions.

Definition 2.1. A nondeterministic finite two-way automaton (2NFA) is given by a quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is the finite set of states, Σ is the finite input alphabet, $\delta : Q \times \Sigma \rightarrow 2^{Q \times \{-1, +1\}}$ is the transition function, q_0 is the starting state, and F is the finite set of accepting states. M is called deterministic (2DFA), if $|\delta(q, a)| \leq 1$ for every $q \in Q$, $a \in \Sigma$. M is a one-way automaton (1NFA, resp., 1DFA) if $\delta(q, a) \subseteq Q \times \{+1\}$ for every $q \in Q$, $a \in \Sigma$. The size $|M|$ of a 2NFA M is given by $|Q|$.

Definition 2.2. A pair $(q, j) \in Q \times \mathbb{N}$ is called configuration, a finite sequence of configurations is called computation. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a 2NFA and $w = a_0 \dots a_{n-1}$ an input. A computation $(p_0, j_0), \dots, (p_m, j_m)$ is called computation of M on w if

- $p_0 = q_0$
- $j_0 = 0, j_m \leq n$
- $\forall i \in \{0, \dots, m-1\} : 0 \leq j_i < n$ and
 $\exists (p, k) \in \delta(p_i, a_{j_i}) : p_{i+1} = p, j_{i+1} = j_i + k$

If $j_m = n$, $p_m \in F$, the computation is called **accepting**. We define

$$L(M) = \{w \in \Sigma^* : \text{there exists an accepting computation of } M \text{ on } w\}.$$

Remarks 2.3. (1) We consider deterministic automata with partial functions as transition functions. Some authors (e.g., [10, 11]) require total transition functions. In particular, there are no so-called TRAP-states in our automata.

- (2) A 2NFA does not possess endmarkers. Several authors (e.g., [6, 4]) consider automata having $\langle w \rangle$ on the input tape, where w is the actual input. The transition function δ is then defined on $Q \times (\Sigma \cup \{<, >\})$, the symbols $<, >$ are part of the machine.
- (3) In order for an input to be accepted, the computation has to leave the input portion of the input tape off the rightmost symbol. In particular, a computation is not accepting if the computation loops on the input.

2.2. Regularity. If A is a class of automata, one can define the class $L(A)$ of languages accepted by automata from A by

$$L(A) = \{L(M) \mid M \in A\}.$$

Now, the class Reg of regular languages is given by

$$Reg = L(1DFA).$$

However, one gets the same class of languages for every other class of automata introduced in the last subsection, that is,

$$(1) \quad L(1DFA) = L(1NFA) = L(2DFA) = L(2NFA).$$

By definition, the following inclusions hold:

$$L(1DFA) \subseteq \frac{L(1NFA)}{L(2DFA)} \subseteq L(2NFA).$$

Thus, in order to obtain (1), for any given $2NFA$ M_1 one has to construct a $1DFA$ M_2 with $L(M_1) = L(M_2)$. This is possible; see Vardi [18] (see also [11]). However, there were earlier equivalence results which we list for completeness:

- Rabin-Scott [14] $2DFA \mapsto 1NFA$
- Shepherdson [16] $2DFA \mapsto 1DFA$
- Vardi [18] $2NFA \mapsto 1DFA$

Rabin-Scott employed a crossing sequence analysis in order to eliminate two-way motion. This construction has the disadvantage that a former deterministic automaton is in general transformed into a nondeterministic automaton. The Shepherdson construction, on the other hand, does not introduce nondeterminism but is still not able to handle both two-way motion and nondeterminism. Vardi generalized the Shepherdson construction to automata from $2NFA$.

2.3. State complexity: issues and concepts. Given a regular language L and two classes of computing devices A_1, A_2 , where $A_1 \subseteq A_2$, one can ask the natural question whether the additional power of the class A_2 results in a description of the particular language L with fewer states. Furthermore, the maximum of this trade-off (taken over all regular languages which can be described by a finite subclass of automata from either A_1 or A_2) is an interesting object. In a slightly more general context, the definitions below provide upper and lower bounds for this quantity.

Definition 2.4 (upper bound). *Let A_1, A_2 be two classes of automata, both accepting exactly the class of regular languages. For monotonically increasing functions $f : \mathbb{N} \rightarrow \mathbb{N}$, $g : \mathbb{N} \rightarrow \mathbb{N}$ we have*

$$\frac{A_1}{f(n)} \rightarrow \frac{A_2}{g(n)} ,$$

if and only if for every $n \in \mathbb{N}$ the following holds: for every $M_1 \in A_1$ having $f(n)$ states, there exists $M_2 \in A_2$ obeying $L(M_1) = L(M_2)$ and $|M_2| \leq g(n)$.

Definition 2.5 (lower bound). *Let A_1, A_2 be two classes of automata, both accepting exactly the class of regular languages. For monotonically increasing functions $f : \mathbb{N} \rightarrow \mathbb{N}$, $g : \mathbb{N} \rightarrow \mathbb{N}$ we have*

$$\frac{A_1}{f(n)} \rightarrow \frac{A_2}{g(n)} ,$$

if and only if for infinitely many $n \in \mathbf{N}$ the following holds: there exists a regular language L_n such that there is an automaton $M_1 \in A_1$ having $f(n)$ states and accepting L_n , and every automaton $M_2 \in A_2$ obeying $L(M_2) = L_n$ has at least $g(n)$ states.

We will mostly consider the case $f = id$.

The constructions from the last subsection yield the following upper bounds:

$$\begin{array}{lll} \text{Rabin} - \text{Scott} & : & \begin{array}{l} 2DFA \rightarrow 1NFA \\ n \leq n^{2n-1} \end{array} \\ \text{Shepherdson} & : & \begin{array}{l} 2DFA \rightarrow 1DFA \\ n \leq (n+1)^{n+1} \end{array} \\ \text{Vardi} & : & \begin{array}{l} 2NFA \rightarrow 1DFA \\ n \leq (|\Sigma| + 1) \cdot 2^{n^2+n} \end{array} \end{array}$$

The constructions of Shepherdson and Vardi were improved by Birget in [4], yielding the following upper bounds:

$$\begin{array}{ll} 2DFA & \rightarrow 1DFA \\ n & \leq n^n \\ 2NFA & \rightarrow 1DFA \\ n & \leq 2^{(n-1)^2+n} \end{array}$$

There are several lower bound results which more or less demonstrate optimality of these constructions.

Theorem 2.6 (Meyer-Fischer '71).

$$\begin{array}{ll} 2DFA & \rightarrow 1DFA \\ 5n + 5 & \geq n^n \end{array}$$

This trade-off is obtained on the sequence

$$L_n = \{0^{i_1}10^{i_2}1 \dots 10^{i_n}2^k0^{i_k} : 1 \leq k \leq n, 1 \leq i_j \leq n\}$$

and yields asymptotic optimality of the Shepherdson-construction. The next section shows an application of this bound and in particular motivates further study of exact sharpness of the upper bound.

Theorem 2.7 (Sakoda-Sipser '78).

$$\begin{array}{ll} 2NFA & \rightarrow 1DFA \\ n & \geq \frac{1}{2}2^{(n-2)^2} \end{array}$$

Definition 2.8. A 2DFA is called **Sweeping automaton (SA)**, if the reading direction is only changed at the leftmost or rightmost symbol of the input.

Theorem 2.9 (Sipser '79).

$$\begin{array}{ll} 1NFA & \rightarrow SA \\ n & \geq 2^n \end{array}$$

Theorem 2.10 (Micali '81).

$$\begin{array}{ll} 2DFA & \rightarrow SA \\ n & \geq c2^n \end{array}$$

We want to add the following lower bound to this list. It will play a major role throughout the paper, in particular in the study of the sharpness of our lower bound for two-way spectra.

Theorem 2.11. *There exists a constant $c > 0$ such that*

$$\begin{array}{ccc} 2DFA & \rightarrow & 1NFA \\ n & & \geq 2^{cn} \end{array}$$

Proof. Consider the following languages

$$L_n = \{awbwa : w \in \{0,1\}^*, |w| = n\}.$$

It is known that every $1NFA$ accepting L_n has at least 2^n states [19]. The $2DFA$ M given in Figure 1 has $\mathcal{O}(n)$ states and accepts L_n . Moreover, it is quite easy to see that an accepting computation has $\mathcal{O}(n^2)$ left moves. \square

3. THE TWO-WAY SPECTRUM OF A REGULAR LANGUAGE L

3.1. Definition and computability.

Definition 3.1. *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a $2DFA$ and $w = a_0 \dots a_{n-1} \in \Sigma^*$. Let $(q_0, i_0), \dots, (q_m, i_m)$ be the corresponding sequence of configurations in M . Define $\lambda(w) = \#\{j : 0 \leq j \leq m, i_j = -1\}$ and $\lambda(M) = \sup\{\lambda(w) : w \in L(M)\}$.*

Definition 3.2. *Define for $k \in \mathbf{N} \cup \{\infty\}$,*

$$2DFA(k) = \{M : M \text{ is a } 2DFA \text{ with } \lambda(M) \leq k\}.$$

Definition 3.3. *The two-way spectrum of L is defined by*

$$\sigma(L) = (\sigma_0(L), \sigma_1(L), \dots; \sigma_\infty(L)),$$

where

$$\sigma_k(L) = \min\{|M| : M \in 2DFA(k), L(M) = L\}.$$

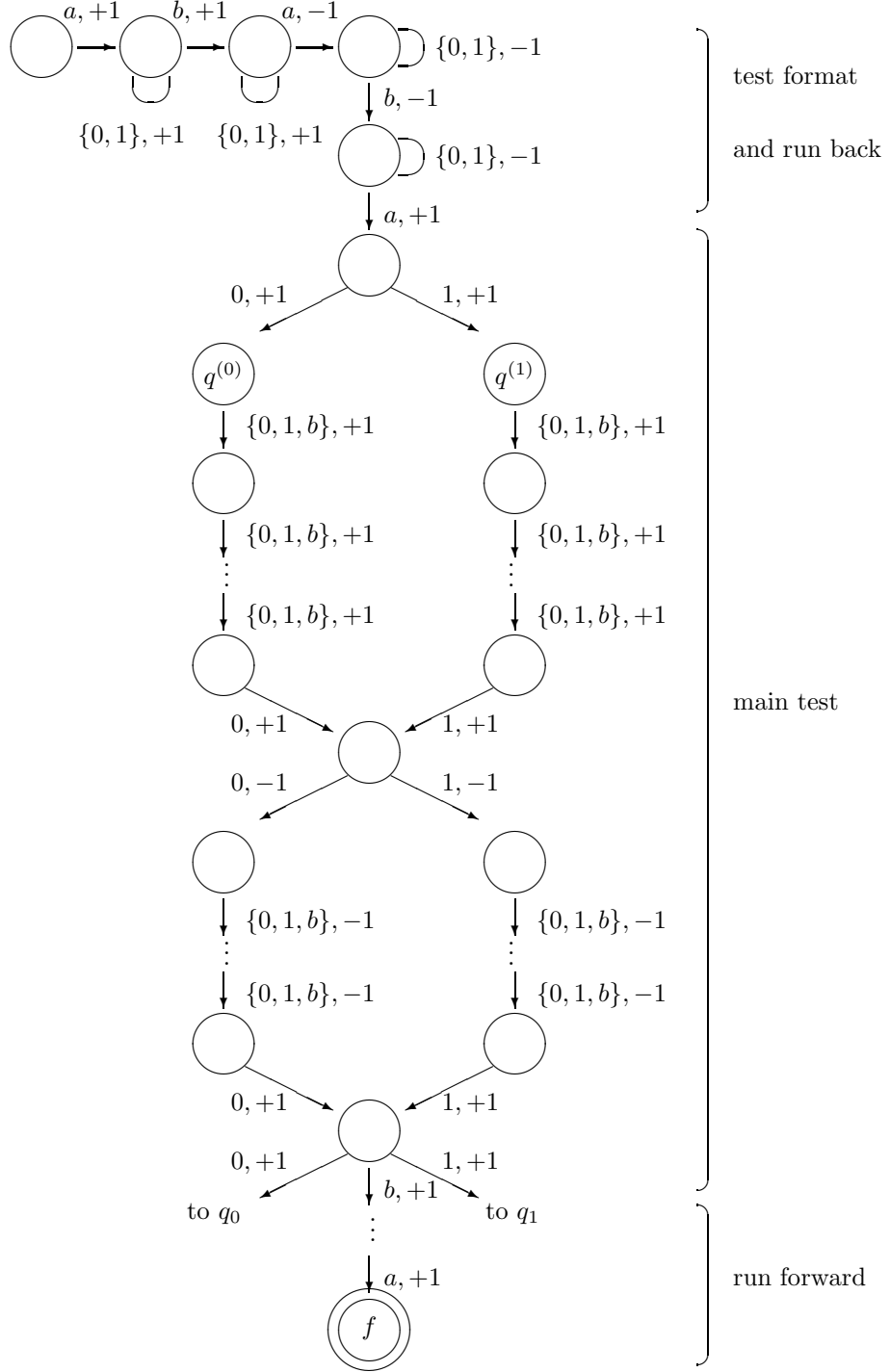
Remark 3.4. *The sequence $\sigma_k(L)$ is monotonically decreasing since $2DFA(k_1) \subseteq 2DFA(k_2)$ if $k_1 \leq k_2$.*

Lemma 3.5. *Let M be a $2DFA$. $\lambda(M) < \infty$ is decidable.*

Proof. Recall the concept of a crossing sequence (see [11]): Write, for a computation of M on some word w , the list of states M is currently in during the computation at hand below each boundary between two consecutive input symbols. This list is called a crossing sequence. It is clear that the first time a boundary is crossed, the head must be moving right. Subsequent crossings must be in alternate directions. If the input w is accepted, it follows that all crossing sequences below the input are of odd length and that no two odd- and no two even-numbered elements in one of these crossing sequences are identical. Call crossing sequences with these two properties *valid*. Thus, if n is the number of states of M , the length of a valid crossing sequence is bounded by an effective constant $B = B(n)$ and the number of valid crossing sequences is bounded by an effective constant $C = C(n)$.

We claim that there is an effective constant $D = D(n)$ such that $\lambda(M) = \infty$ if and only if there is a word $w \in L(M)$ having length bounded by D such that the accepting computation of M on w has a repeated crossing sequence which contains at least three elements. This condition is decidable.

Consider first an accepting computation of M on a word $w = w_1 \dots w_n$ such that the crossing sequences right of w_i and right of w_k coincide and contain at least three

FIGURE 1. Automaton M from the proof of Theorem 2.11

elements. Then we can consider the words $w^{(l)} = w_1 \dots w_i (w_{i+1} \dots w_k)^l w_{k+1} \dots w_n$ which all belong to $L(M)$. For every $N \in \mathbf{N}$, there is l such that $\lambda(w^{(l)}) > N$. Hence, $\lambda(M) = \infty$.

Conversely, let $\lambda(M) = \infty$. Since the length of valid crossing sequences is uniformly bounded by $B(n)$, we can find a word $w \in L(M)$ such that the accepting computation of M on w has a repeated crossing sequence which has at least three elements. While the argument above used a pumping argument, we shall now employ an inverse-pumping argument to prune the word w in order to obtain a word $w' \in L(M)$ which still has a repeated crossing sequence with at least three elements and whose length we can bound by an effective constant depending only on n . First, one can delete parts from w until exactly one repeated crossing sequence is left. Note that this yields an accepting computation. Next, one deletes parts from the stretches where there are consecutive crossing sequences of length one so that there are no repeated states left on such stretches. This will again yield an accepting computation. The resulting word $w' \in L(M)$ will have its length bounded by $n(2 + C(n))$. \square

Lemma 3.6. *Let M be a 2DFA obeying $\lambda(M) < \infty$ and let $k \in \mathbf{N}$. $\lambda(M) = k$ is decidable.*

Proof. Let $T_k(M) = \{w : w \in L(M), \lambda(w) \leq k\}$. Obviously,

$$\lambda(M) = k \Leftrightarrow T_{k-1}(M) \neq T_k(M) = L(M).$$

We are therefore done with the proof if we show that $T_k(M)$ is regular. The idea is the following: Count the left moves in a homomorphic image and restrict their number by k . We introduce the following alphabet:

$$\Sigma_T = \{[p, a, r, q] : \delta(p, a) = (q, r)\}.$$

Let $R \subseteq \Sigma_T^*$ be the (regular) set of accepting computations of M :

$$R = \{[p_0, a_0, r_0, p_1][p_1, a_1, r_1, p_2] \dots [p_{n-1}, a_{n-1}, r_{n-1}, p_n] :$$

$$\delta(p_i, a_i) = (p_{i+1}, r_i), p_0 = q_0, p_n \in F, r_0 = r_{n-1} = +1\} \cup \{\varepsilon : q_0 \in F\}$$

Let $S_k = \{1^i : 0 \leq i \leq k\}$ and

$$h : \Sigma_T^* \rightarrow \{1\}^*, [p, a, r, q] \mapsto \begin{cases} 1, & r = -1 \\ \varepsilon, & r = +1 \end{cases}.$$

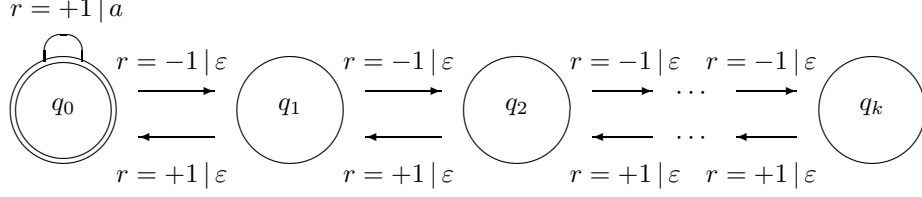
With the finite transducer E_k from Figure 2, we have $T_k(M) = E_k(R \cap h^{-1}(S_k))$. Thus, $T_k(M)$ is regular. \square

Theorem 3.7. *$\sigma(L)$ is computable.*

Proof. Let M be the minimal 1DFA with $L(M) = L$. Then, $\sigma_0(L) = |M|$. Let M_1, \dots, M_n be the (finitely many and effectively determinable) 2DFA obeying $|M_i| \leq \sigma_0(L)$ and $L(M_i) = L$. Compute the complexities $\lambda(M_i)$, using Lemma 3.5 and Lemma 3.6. Then, determine minimal pairs $(|M_i|, \lambda(M_i))$, where minimality is understood according to the order $(x_1, y_1) \leq (x_2, y_2) :\Leftrightarrow x_1 \leq x_2, y_1 \leq y_2$. This gives the sequence

$$(s_1, \lambda_1), \dots, (s_j, \lambda_j)$$

of minimal pairs obeying $s_1 > \dots > s_j$ and $0 = \lambda_1 < \dots < \lambda_j$. Let $\lambda_{j+1} = \infty$. Then, $\sigma_k(L) = s_l \forall \lambda_l \leq k < \lambda_{l+1}$ and $\sigma_\infty(L) = s_j$. \square

FIGURE 2. Transducer E_k

3.2. Examples.

3.2.1. *The constant spectrum.* The sequence of languages $L_n = \{1^{n-1}\} \subseteq 1^*$, $n \in \mathbf{N}$, exhibits the occurrence of a constant spectrum $(n, n, \dots; n)$. One checks (cf. [5]):

- $\sigma_0(L_n) \leq n$,
- $\sigma_\infty(L_n) \geq n$.

3.2.2. *The collapse at $n \in \mathbf{N}$.* We aim at the construction of a spectrum which is constant for $\{k \mid k = 0, \dots, n-1\}$ and for $\{k \mid k = n, n+1, \dots, \infty\}$, and which has a major jump in the transition from $n-1$ to n . Consider the languages $L_n = \{0, 1\}^* 1 \{0, 1\}^{n-1} \$$, $n \in \mathbf{N}$. The sequence of lemmas below establishes the desired form of the spectrum.

Lemma 3.8. $\sigma_0(L_n) \leq 2^n + 1$.

Proof. Define the 1DFA $M = (Q, \Sigma, \delta, q_0, F)$ by $Q = \{0, 1\}^n \cup \{f\}$, $\Sigma = \{0, 1, \$\}$, $q_0 = (0, \dots, 0)$, $F = \{f\}$ and

$$\delta((a_1, \dots, a_n), a) = \begin{cases} (a_2, \dots, a_n, a) & a \in \{0, 1\} \\ f & \text{if } a = \$ \text{ and } a_1 = 1 \\ \text{undefined} & \text{else} \end{cases}$$

Obviously, $L(M) = L_n$ and $|M| = 2^n + 1$. □

Lemma 3.9. $\sigma_{n-1}(L_n) \geq 2^n + 1$.

Proof. Let $M = (Q, \{0, 1, \$\}, \delta, q_0, F) \in 2DFA(n-1)$ obey $L(M) = L_n$. Then, by $\lambda(M) \leq n-1$,

$$\exists \hat{w} : \lambda(\hat{w}) = \max\{\lambda(v) : v \in \{0, 1\}^*\}.$$

M eventually leaves \hat{w} off its rightmost symbol and, by definition of \hat{w} , M then moves to the right until $\$$ is read. Now, every $w \in \{0, 1\}^*$ is prefix of a word in L_n . Thus, the computation of M on w reaches the rightmost symbol of w in a state q_w . We want to show that $\{\hat{w}v \mid v \in \{0, 1\}^n\}$ is a set of words that distinguishes between states. Suppose to the contrary

$$q_{\hat{w}v_1} = q_{\hat{w}v_2} \text{ for } v_1, v_2 \in \{0, 1\}^n, v_1 \neq v_2.$$

Without loss of generality, $v_1 = x_1 1 y$, $v_2 = x_2 0 y$ with $|x_1| = |x_2|$. $\hat{w}v_1 0^{n-|y|-1} \$$ and $\hat{w}v_2 0^{n-|y|-1} \$$ are therefore either both accepted or both rejected, contradiction.

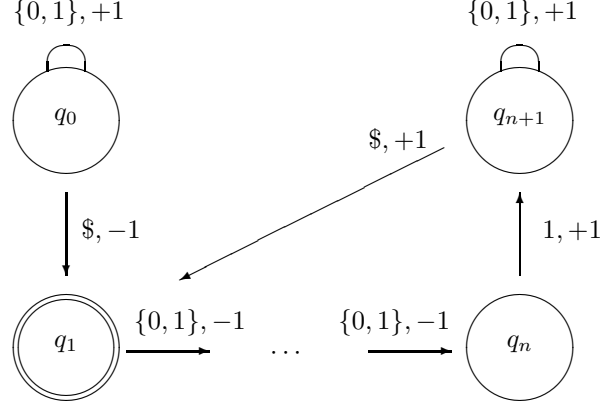


FIGURE 3. Automaton from the proof of Lemma 3.10

Hence, $|Q| \geq 2^n$. Since $L_n \neq \emptyset$ and all the $q_{\bar{w}v}$ are non-accepting, we conclude $|Q| \geq 2^n + 1$. \square

Lemma 3.10. $\sigma_n(L_n) \leq n + 2$.

Proof. Consider the automaton in Figure 3. One easily checks that this automaton accepts L_n and belongs to $2DFA(n)$. \square

Lemma 3.11. $\sigma_\infty(L_n) \geq n + 2$.

Proof. When a $2DFA$ M reaches ' $\$$ ', M has to move n symbols backwards, since otherwise $|M| \geq 2^n + 1$ (compare the proof of Lemma 3.9). In order to count n arbitrary symbols, a $2DFA$ needs $n + 1$ states [5]. In those states (with the exception of the last $(1, +1)$ -transition) only left moves are defined for inputs $0, 1$. Thus, M has at least one more state, since L_n is not empty. \square

3.2.3. *The collapse at ∞ .* Consider the Meyer-Fischer languages

$$L_n = \{0^{i_1}10^{i_2}1 \dots 10^{i_n}2^m0^{i_m} \mid 1 \leq m \leq n, 1 \leq i_j \leq n\},$$

which yield asymptotic optimality of the Shepherdson-construction. In the next subsection it will be shown that for $L'_n = (\$L_n\$)^*$ the two-way complexity collapses at infinity:

- $\sigma_k(L'_n) = \Omega(n^n)$ for every $k \in \mathbb{N}$,
- $\sigma_\infty(L'_n) = \mathcal{O}(n)$.

3.2.4. *The decreasing spectrum.* The languages

$$R_n = c\{awbwa : w \in \{0, 1\}^1\}c \dots c\{awbwa : w \in \{0, 1\}^n\}c$$

have spectra with many points of decrease. The idea is the following: The more two-way motion is allowed, the more blocks can be tested in two-way fashion. This argument will be formalized at the end of this section.

3.3. Upper bounds for $\sigma(L)$. The Birget improvement of the Shepherdson construction immediately yields

Theorem 3.12. *Let $L \subseteq \Sigma^*$ be regular with $n = \sigma_\infty(L)$. Then,*

$$\sigma(L) \leq (n^n, n^n, \dots; n).$$

Remark 3.13. *Fixing the spectrum at $\sigma_0(L) = n$ yields the trivial upper bound $(n, n, \dots; n)$, which is sharp by Example 3.2.1.*

In order to study sharpness of the bound from Theorem 3.12, we have to look for languages L for which $\sigma(L)$ is constant with the exception of $\sigma_\infty(L)$. The following lemma gives a class of such languages.

Lemma 3.14. *Let $L \subseteq \Sigma^*$ be regular and $\$ \notin \Sigma$. Then, there exist $k, n \in \mathbf{N}$ with*

$$\sigma((\$L\$)^*) = (k, k, \dots; n).$$

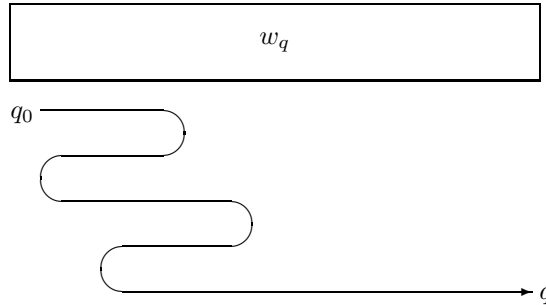
Proof. Let $L \subseteq \Sigma^*$ be an arbitrary regular language. Define $L' = (\$L\$)^*$. We will freely use the following simple properties

$$\begin{aligned} v, w \in L' &\Rightarrow vw \in L', \\ vw \in L', v \in L' &\Rightarrow w \in L', \\ vw \in L', w \in L' &\Rightarrow v \in L'. \end{aligned}$$

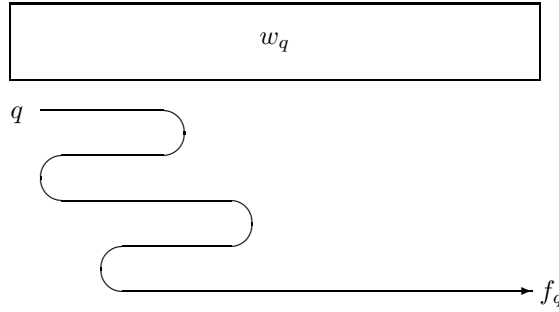
It suffices to show

$$\sigma_k(L') \geq \sigma_0(L') \quad \forall k \in \mathbf{N}.$$

To a given $M = (Q, \Sigma \cup \{\$\}, \delta, q_0, F) \in 2DFA(k)$ with $L(M) = L'$ we will construct an equivalent $2DFA(0)$ $N = (Q, \Sigma \cup \{\$\}, \delta', q'_0, F')$ with $L(N) = L'$. In particular, we have $|N| = |M|$. Define the sets S and T as follows. The set S consists of the states $q \in Q$ with the following property: There exists a word $w_q \in L'$ such that M runs on w_q from q_0 to q :



The set T consists of the states $q \in Q$ with the following property: There exist $w_q \in L'$ and $f_q \in F$ such that:



The transition function δ' results from δ by omitting left moves. Thus δ' is the transition function of a one-way automaton and it makes sense to write $\delta'(P, W)$ for some subset P of Q and a set of words W . Below we will show

Claim 1. $L' = \{w \mid \delta'(S, w) \cap T \neq \emptyset\}$.

Claim 2. Let S_0 be a minimal subset of S which satisfies Claim 1. Let $s_0 \in S_0$ be arbitrary. Then, $L' = \{w \mid \delta'(s_0, w) \cap T \neq \emptyset\}$.

Choosing $q'_0 = s_0 \in S_0$ arbitrarily and defining $F' = T$, we get the desired automaton $N \in 2DFA(0)$.

Proof of Claim 1. ‘ \subseteq ’ Let $v \in L'$. Then, $v^{k+1} \in L'$. Since $\lambda(M) \leq k$ there is j such that there is no left move on the j -th v -block. Let q_1 (resp., q_2) be the state M is in as the computation enters (resp., exists) this v -block. Then, $q_1 \in S$, $q_2 \in T$, and $\delta'(q_1, v) = q_2$. Thus, $v \in \{w \mid \delta'(S, w) \cap T \neq \emptyset\}$.

' \supseteq ' Let $v \in \{w \mid \delta'(S, w) \cap T \neq \emptyset\}$. By definition of S and T , there exist $x, y \in L'$ such that $xyw \in L'$. Hence, $w \in L'$.

Proof of Claim 2. By Claim 1 we only have to show that $w \in L'$ implies $\delta'(s_0, w) \cap T \neq \emptyset$. Suppose there is $w \in L'$ with $\delta'(s_0, w) \cap T = \emptyset$. Then,

$$(2) \quad \delta'(s_0, wL') \cap T = \emptyset.$$

Since S_0 satisfies Claim 1, we have

$$(3) \quad \delta'(S_0, wv) \cap T \neq \emptyset \quad \forall v \in L'.$$

(2) and (3) imply $\delta'(S_0 \setminus \{s_0\}, wv) \cap T \neq \emptyset \quad \forall v \in L'$. Let $\hat{S}_0 = \delta'(S_0 \setminus \{s_0\}, w)$. Then, $\hat{S}_0 \subseteq \delta'(S, L') \subseteq S$ and $\delta'(\hat{S}_0, v) \cap T \neq \emptyset \quad \forall v \in L'$, and we get the chain of inclusions

$$L' \subseteq \{v \mid \delta'(\hat{S}_0, v) \cap T \neq \emptyset\} \subseteq \{v \mid \delta'(S, v) \cap T \neq \emptyset\} = L'.$$

Thus, equality holds. In particular, \hat{S}_0 satisfies Claim 1. By $\#\hat{S}_0 \leq \#(S_0 \setminus \{s_0\}) < \#S_0$, this contradicts the minimality of S_0 . \square

Lemma 3.15. *Let $L \subseteq \Sigma^*$ be regular and $\sigma_0(L) \geq f(\sigma_\infty(L))$ with a monotonically increasing function f . Let $L' = (\$L\$)^*$. Then,*

$$\sigma_0(L') \geq f(\sigma_\infty(L') - 1) + 1.$$

Proof. By adding one state, the minimal automata for L can be modified to yield automata for L' . For $k = 0$, one-way motion is preserved. We therefore have $\sigma_0(L') \leq \sigma_0(L) + 1$ and $\sigma_\infty(L') \leq \sigma_\infty(L) + 1$. We will now show

$$(4) \quad \sigma_0(L') \geq \sigma_0(L) + 1,$$

yielding

$$\sigma_0(L') = \sigma_0(L) + 1 \geq f(\sigma_\infty(L)) + 1 \geq f(\sigma_\infty(L') - 1) + 1.$$

Proof of (4): We will use the fact that for regular $L \subseteq \Sigma^*$, $\sigma_0(L)$ is equal to the number of non-empty left quotients $w \setminus L$, $w \in \Sigma^*$.

For every w not containing $\$$, we have $(\$w) \setminus (\$L\$)^* = w \setminus L\$ (\$L\$)^*$. Thus, L' has at least as many non-empty left quotients as L . Since $L' = \varepsilon \setminus L'$ is another one, L' has at least one more. \square

Theorem 3.16. *Let L_n be regular and f monotonically increasing with $\sigma_\infty(L_n) = n$ and $\sigma_0(L_n) \geq f(n)$. Then, there exists a sequence of regular languages L'_n with*

$$\sigma(L'_n) \geq (f(n-1) + 1, f(n-1) + 1, \dots; n)$$

for infinitely many n .

Proof. The assertion follows from Lemmas 3.14 and 3.15. \square

Remark 3.17. *We have reduced sharpness issues for the upper bound for $\sigma(L)$ given in Theorem 3.12 to the problem*

$$\begin{array}{ccc} 2DFA & \rightarrow & 1DFA \\ n & & \geq ? \end{array}$$

that is, the question to what extent the Shepherdson construction is sharp.

3.4. Lower bounds for $\sigma(L)$. We now aim at proving a similar uniform lower bound for two-way spectra. We first focus on the following upper bound. By reversing the view we will deduce the desired lower bound below.

$$\begin{array}{ccc} 2DFA(k) & \rightarrow & 1DFA \\ n & & \leq ? \end{array}.$$

Let M be a $2DFA(k)$ over Σ . Intuitively, we can construct an equivalent $1DFA$ M' in two steps as follows:

Step 1: Store the last k symbols in the finite control. Simulate left-moves in M by using ε -moves.

Step 2: Eliminate ε -moves.

Theorem 3.18. *For $k \in \mathbf{N}$ we have:*

$$\begin{array}{ccc} 2DFA(k) & \rightarrow & 1DFA \\ n & & \leq n \cdot (k+1) \cdot (|\Sigma| + 1)^{k+1} \end{array}$$

Remarks 3.19. (1) *The trade-off is linear if k and $|\Sigma|$ are fixed. The analogue in the nondeterminism case does not hold [7]!*

(2) *The trade-off is exponential in k .*

(3) *Compared with the Shepherdson-construction, our method has no advantage for $k \geq \Omega(n \cdot \ln n)$.*

Proof. We formalize the two steps from above. Let $M = (Q, \Sigma, \delta, q_0, F) \in 2DFA(k)$.

Step 1 : Construction of an equivalent $1DFA$ $M' = (Q', \Sigma, \delta', q'_0, F')$ with $|Q'| = |Q| \cdot (k+1) \cdot (|\Sigma| + 1)^{k+1}$

Let $\$ \notin \Sigma$. Define $Q' = Q \times \{0, \dots, k\} \times (\Sigma \cup \{\$\})^{k+1}$, $q'_0 = (q_0, 0, \$, \dots, \$)$, $F' = \{(f, 0, a_k, \dots, a_1, \$) : f \in F; a_i \in \Sigma \cup \{\$\}, 1 \leq i \leq k\}$ and

$$\begin{aligned}
\delta'((p, j, a_k, \dots, a_0), \varepsilon) &= \begin{cases} (q, j-1, a_k, \dots, a_0) & \text{if } \delta(p, a_j) = (q, +1), j > 0 \\ (q, j+1, a_k, \dots, a_0) & \text{if } \delta(p, a_j) = (q, -1), j > 0 \end{cases} \\
\delta'((p, 0, a_k, \dots, a_1, a_0), \varepsilon) &= \begin{cases} (q, 0, a_{k-1}, \dots, a_0, \$) & \text{if } \delta(p, a_0) = (q, +1) \\ (q, 1, a_k, \dots, a_0) & \text{if } \delta(p, a_0) = (q, -1) \end{cases} \\
\delta'((p, 0, a_k, \dots, a_1, \$), a) &= \begin{cases} (q, 0, a_{k-1}, \dots, a_1, a, \$) & \text{if } \delta(p, a) = (q, +1) \\ (q, 1, a_k, \dots, a_1, a) & \text{if } \delta(p, a) = (q, -1) \end{cases}
\end{aligned}$$

Obviously, M' is deterministic and satisfies $L(M') = L(M)$.

Step 2 : Elimination of ε -moves

Define $M'' = (Q', \Sigma, \delta'', q'_0, F'') \in 1DFA$, where

$$\begin{aligned}
\varepsilon(q) &= \{q' : \delta'(q, \varepsilon^*) = q' \text{ and } \delta'(q', \varepsilon) \text{ is undefined}\} \\
\delta''(q, a) &= \begin{cases} \delta'(q, a) & \text{if } \delta'(q, a) \neq \emptyset \\ \delta'(\varepsilon(q), a) & \text{if } \varepsilon(q) \neq \emptyset \end{cases} \\
F'' &= F' \cup \{q'_0 : \varepsilon(q'_0) \cap F' \neq \emptyset\}
\end{aligned}$$

δ'' is well-defined and M'' has $|Q| \cdot (k+1) \cdot (|\Sigma| + 1)^{k+1}$ states. \square

Corollary 3.20. *For $k \in \mathbb{N}$ we have*

$$\begin{aligned}
1DFA &\rightarrow 2DFA(k) \\
n &\geq n \cdot (k+1)^{-1} \cdot (|\Sigma| + 1)^{-(k+1)}
\end{aligned}$$

This yields the following lower bound for $\sigma(L)$:

Theorem 3.21. *Let $L \subseteq \Sigma^*$ be regular with $\sigma_0(L) \geq n^n$. Then,*

$$\sigma(L) \geq (n^n, \frac{1}{2}c^2 \cdot n^n, \frac{1}{3}c^3 \cdot n^n, \frac{1}{4}c^4 \cdot n^n, \dots, n, n, \dots; n),$$

where $c = (|\Sigma| + 1)^{-1}$.

Remark 3.22. *Fixing the other end of the spectrum again yields the trivial and sharp lower bound $(n, n, \dots; n)$.*

We are going to study the sharpness of the bound from Theorem 3.21. To this end, we will follow a strategy which relies on a concatenation procedure. We will consider languages which have a block structure. On each block, the use of two-way motion will imply savings of state complexity. The following theorem shows that, for the class $1DFA$, the state complexities of the blocks actually add up. This yields unrestricted freedom in the choice of the block languages.

Theorem 3.23 (Concatenation Lemma). *Let L_1, \dots, L_n be non-empty languages over Σ . Let*

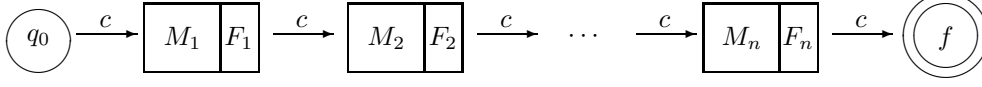
$$L = cL_1cL_2c \dots cL_nc,$$

where $c \notin \Sigma$. Then,

$$\sigma_0(L) = 2 + \sum_{i=1}^n \sigma_0(L_i).$$

Proof. Let M_1, \dots, M_n be the minimal $1DFA$'s with $L(M_i) = L_i$, where

- $M_i = (Q, \Sigma, \delta_i, q_{0,i}, F_i)$,
- $Q_i \cap Q_j = \emptyset$ for $i \neq j$ (wlog).

FIGURE 4. Automaton M from the proof of Theorem 3.23

Define $M = (Q, \Sigma \cup \{c\}, \delta, q_0, F)$ by (cf. Figure 4)

$$\begin{aligned} Q &= \{q_0, f\} \cup Q_1 \cup \dots \cup Q_n \\ F &= \{f\} \\ \delta(p, a) &= \begin{cases} \delta_i(p, a) & \text{if } p \in Q_i, a \in \Sigma \\ q_{0,i+1} & \text{if } p \in F_i, 1 \leq i < n, a = c \\ q_{0,1} & \text{if } p = q_0, a = c \\ f & \text{if } p \in F_n, a = c \end{cases} \end{aligned}$$

Since M has the desired number of states we are done with the proof if we can show:

(1) $L(M) = L$.

(2) M is minimal.

(1) is immediate from the construction.

(2) requires that

- every state in M is reachable.
- for every $p \neq q$ there exists $w_{p,q}$ with $\delta(p, w_{p,q}) = f \Leftrightarrow \delta(q, w_{p,q}) \neq f$.

Reachability follows from $L_i \neq \emptyset$ and reachability in every M_i . Let $p, q \in Q$ with $p \neq q$. We consider the following cases:

Case 1: one of the two states is q_0 .

Choose $w_i \in L_i$ and define $w_{p,q} = cw_1c \dots cw_nc$.

Case 2: one of the two states is f .

Let $w_{p,q} = \varepsilon$.

Case 3: both states come from M_i .

Choose $\tilde{w}_{p,q} \in \Sigma^*$ with $\delta_i(p, \tilde{w}_{p,q}) \in F_i \Leftrightarrow \delta_i(q, \tilde{w}_{p,q}) \notin F_i$,

and $w_{i+1} \in L_{i+1}, \dots, w_n \in L_n$. Define $w_{p,q} = \tilde{w}_{p,q}cw_{i+1}c \dots cw_nc$.

Case 4: $p \in M_i, q \in M_j$, wlog $i < j$.

Choose $\tilde{w}_q \in \Sigma^*$ with $\delta_j(q, \tilde{w}_q) \in F_j$ and $w_{j+1} \in L_{j+1}, \dots, w_n \in L_n$.

Define $w_{p,q} = \tilde{w}_qcw_{j+1}c \dots cw_nc$.

One checks that the above choices of the $w_{p,q}$ have the desired property.

This concludes the proof. \square

Consider the languages

$$L_i = \{awbwa : w \in \{0, 1\}^i\}$$

and their concatenation

$$R_n = cL_1c \dots cL_nc.$$

Lemma 3.24. *There exists a 1DFA M_i with $4 \cdot 2^i$ states accepting L_i . M_i is minimal, that is, $\sigma_0(L_i) = 4 \cdot 2^i$.*

Proof. Define the 1DFA $M_i = (Q_i, \{0, 1, a, b\}, \delta_i, q_{0,i}, F_i)$ by

$$\begin{aligned}
Q_i &= \{(x_1, \dots, x_l, \text{flag}) : x_j \in \{0, 1\}, 1 \leq l \leq i, \\
&\quad \text{flag} \in \{\text{read}, \text{compare}\}\} \cup \{q_{0,i}, f_i, q_s, q_e\} \\
F_i &= \{f_i\} \\
\delta(q_{0,i}, a) &= q_s \\
\delta(q_s, x) &= (x, \text{read}) \quad \text{if } x \in \{0, 1\} \\
\delta((x_1, \dots, x_l, \text{read}), x) &= \begin{cases} (x_1, \dots, x_l, x, \text{read}) & 1 \leq l \leq i-1, x \in \{0, 1\} \\ (x_1, \dots, x_l, \text{compare}) & l = i, x = b \\ \text{undefined} & \text{else} \end{cases} \\
\delta((x_1, \dots, x_l, \text{compare}), x) &= \begin{cases} (x_2, \dots, x_l, \text{compare}) & 2 \leq l \leq i, x = x_1 \\ q_e & l = 1, x = x_1 \\ \text{undefined} & \text{else} \end{cases} \\
\delta(q_e, a) &= f_i
\end{aligned}$$

M_i has the claimed number of states:

$$|Q_i| = 2 \cdot \sum_{l=1}^i 2^l + 4 = 2 \cdot (2^{i+1} - 2) + 4 = 2^{i+2}.$$

In order to show that M_i is minimal, we have to show reachability and separability of the set of states. Reachability is seen as follows:

$$\begin{aligned}
q_{0,i} &= \delta(q_{0,i}, \varepsilon) \\
q_s &= \delta(q_{0,i}, a) \\
(x_1, \dots, x_l, \text{read}) &= \delta(q_{0,i}, ax_1 \dots x_l) \\
(x_1, \dots, x_l, \text{compare}) &= \delta(q_{0,i}, a0^{i-l}x_1 \dots x_l b0^{i-l}) \\
q_e &= \delta(q_{0,i}, a0^i b0^i) \\
f &= \delta(q_{0,i}, a0^i b0^i a)
\end{aligned}$$

The separating words are given in the following table:

state	state	separating word
$q_{0,i}$	arbitrary	$a0^i b0^i a$
q_s	arbitrary	$0^i b0^i a$
q_e	arbitrary	a
f	arbitrary	ε
$(x_1, \dots, x_l, \text{read})$	arbitrary	$0^{i-l} b x_1 \dots x_l 0^{i-l} a$
$(x_1, \dots, x_{l_1}, \text{compare})$	$(x'_1, \dots, x'_{l_2}, \text{compare})$	$x_1 \dots x_{l_1} a$

Thus, M_i is minimal. □

Lemma 3.25. *There exists a 2DFA N_i with $\mathcal{O}(i)$ states accepting L_i . We have $k(i) = \lambda(N_i) = \mathcal{O}(i^2)$.*

Proof. The assertion follows from the proof of Theorem 2.11. □

Theorem 3.26. *We have*

$$\sigma_0(R_n) = 2 + 4 \cdot (2^1 + \dots + 2^n).$$

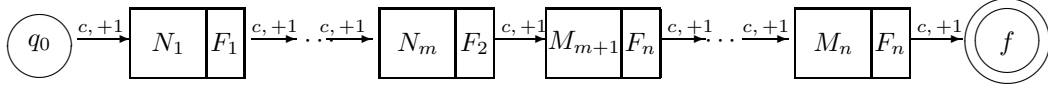
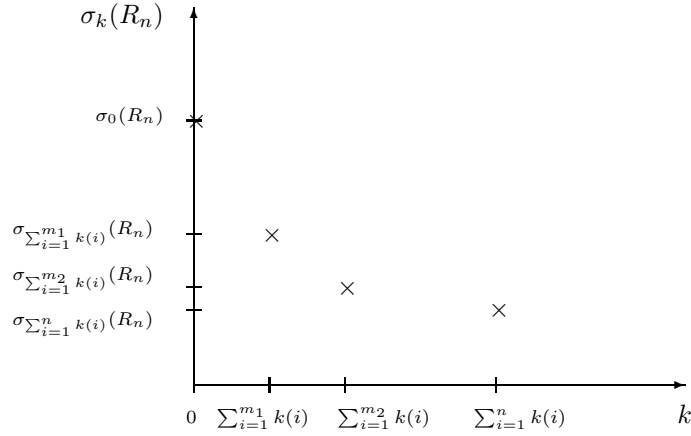
FIGURE 5. Automaton M from the proof of Theorem 3.27

FIGURE 6. Upper bounds established in Theorem 3.27

Proof. Follows from Lemma 3.24 and the Concatenation Lemma. \square

Theorem 3.27. For $1 \leq m \leq n$ we have

$$\sigma_{\sum_{i=1}^m k(i)}(R_n) \leq 2 + c \cdot (1 + \dots + m + 2^{m+1} + \dots + 2^n).$$

In particular:

$$\sigma_{\sum_{i=1}^n k(i)}(R_n) \leq 2 + c \cdot (1 + \dots + n).$$

Proof. The number of allowed left-moves suffices to check the first m blocks in two-way fashion. We therefore can construct the automaton $M \in 2DFA(\sum_{i=1}^m k(i))$ for R_n as given in Figure 5. This yields the desired estimate for $\sigma_{\sum_{i=1}^m k(i)}(R_n)$. \square

Thus, we qualitatively have the picture given by Figure 6.

4. STATE COMPLEXITY THEOREMS FOR THE CLASS $2DFA(k)$

4.1. Comparison with $1NFA$.

Theorem 4.1. For every $k \in \mathbf{N}$, there exists a constant c_k such that

$$\begin{array}{ccc} 1NFA & \rightarrow & 2DFA(k) \\ n & & \geq 2^{c_k \sqrt{n}} \end{array}$$

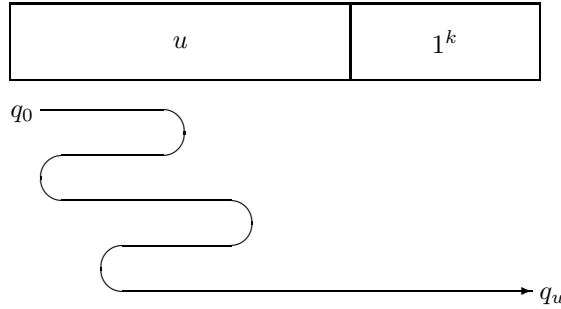
Proof. Consider the regular languages

$$L_{n,k} = \{u1^k v : u, v \in \{0,1\}^n, u \neq v\}.$$

A straightforward modification of the well-known $\mathcal{O}(n^2)$ -state *1NFA* for $\{uv : u, v \in \{0,1\}^n, u \neq v\}$ (cf. [19]) yields a *1NFA* for $L_{n,k}$ with $\mathcal{O}(c'(k) \cdot n^2)$ states. Now, let $M \in 2DFA(k)$ with $L(M) = L_{n,k}$. We have to show that M has at least 2^n states. Consider the words

$$\{u1^k : u \in \{0,1\}^n\}.$$

They are prefixes of words in $L_{n,k}$. Thus, M does not loop on them since M is deterministic. One has the following situation:



The assertion now follows from

$$u_1 \neq u_2 \Rightarrow q_{u_1} \neq q_{u_2},$$

which again holds by determinism. \square

Remark 4.2. *The result can be improved. Using the languages $L_n = \{0,1\}^*1\{0,1\}^{n-1}\$$ we have, by results from Section 3,*

$$\forall n \geq k+1 : \begin{array}{ccc} 1NFA & \rightarrow & 2DFA(k) \\ n+2 & & \geq 2^n + 1 \end{array}$$

This provides an analogue of the result of Sipser: By restriction of two-way motion one obtains an exponential trade-off between nondeterminism and two-way motion.

The following theorem gives a result for the other direction.

Theorem 4.3. *Let $k \in \mathbf{N}$. Then, there exist constants c_1, c_2 , such that*

$$\begin{array}{ccc} 2DFA(k) & \rightarrow & 1NFA \\ n & & \geq \frac{1}{c_1 k} \cdot n \cdot 2^{c_2 \sqrt{k}} \end{array}$$

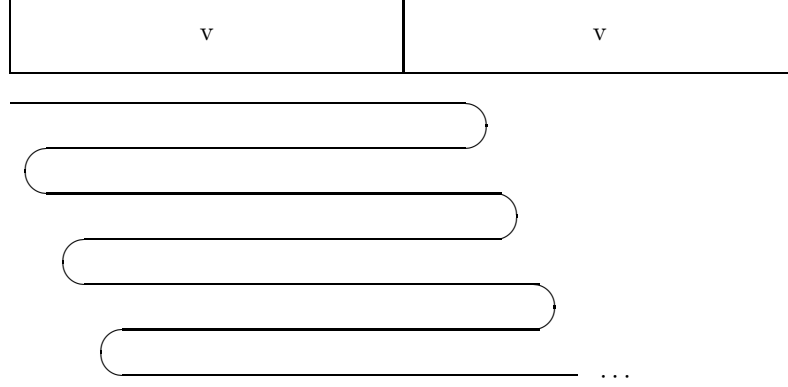
Proof. Consider the languages

$$L_n = \{wvww : wv \in \{0,1\}^n, |v| = \lfloor c \cdot \sqrt{k} \rfloor\}$$

with c to be specified below. In a similar way as in the proof of the corresponding result for

$$L'_n = \{ww : w \in \{0,1\}^n\}$$

as given, for example, in [19], one checks that every *1NFA* accepting L_n has at least 2^n states. Now, a $2DFA(k)$ can use the following strategy: store the first $n - c \cdot \sqrt{k}$ symbols (i.e. w) in the finite control and check the next $2 \cdot c \cdot \sqrt{k}$ symbols by using the k left-moves:



Choose c such that the k left moves suffice to test the middle part, compare the proof of Theorem 2.11. The $2DFA(k)$ is therefore a cartesian product of two automata. We have the following estimate for the number of states:

$$|M| \leq 2^{n-c_2\sqrt{k}} \cdot c_1 k.$$

Thus,

$$|\min. \text{ 1NFA for } L_n| \geq \frac{1}{c_1 k} \cdot 2^{c_2\sqrt{k}} \cdot |\min. \text{ 2DFA}(k) \text{ for } L_n|,$$

concluding the proof. \square

4.2. Comparison with 1DFA and 2DFA. We are now interested in trade-offs coming from eliminating (resp., allowing) two-way motion, starting from the class $2DFA(k)$, that is,

$$2DFA \rightsquigarrow 2DFA(k) \rightsquigarrow 1DFA.$$

Theorem 4.4. *For every $k \in \mathbf{N}$, we have*

$$\begin{array}{ccc} 2DFA & \rightarrow & 2DFA(k) \\ 5(n+1)+5 & \geq & n^n+1 \end{array}$$

This result was already proven in the the previous section.

Theorem 4.5. *There is a constant c such that for $k \in \mathbf{N}$, we have*

$$\begin{array}{ccc} 2DFA(k) & \rightarrow & 1DFA \\ n+k+3 & \geq & c \cdot n \cdot 2^k \end{array}$$

The proof of this theorem will be given in two separate lemmas. We consider the languages

$$L_{n,k} = \{0,1\}^* 1 \{0,1\}^{k-1} \$_1 \cup \{w \in \{0,1\}^* : \#_1(w) = n-1\} \$_2.$$

First, we give a $2DFA(k)$ for $L_{n,k}$.

Lemma 4.6. *For $n, k \in \mathbf{N}$, there is $M \in 2DFA(k)$ with $L(M) = L_{n,k}$ and $|M| = n+k+3$*

Proof. M runs to the rightmost symbol of the input while counting the 1's. In the case that the rightmost symbol is $\$2$, M accepts if and only if $n - 1$ 1's were read. If the rightmost symbol is $\$1$, M runs back k symbols and checks for a 1. Formally, let $M = (Q, \{0, 1, \$1, \$2\}, \delta, q_0, F)$ be defined by:

$$\begin{aligned}
Q &= \{q_0, \dots, q_{n+k+1}, f\} \\
F &= \{f\} \\
\delta(q_i, 0) &= (q_i, +1) \text{ if } 0 \leq i \leq n \\
\delta(q_i, 1) &= (q_{i+1}, +1) \text{ if } 0 \leq i \leq n - 1 \\
\delta(q_n, 1) &= (q_n, +1) \\
\delta(q_i, \$1) &= (q_{n+1}, -1) \text{ if } 0 \leq i \leq n \\
\delta(q_{n-1}, \$2) &= (f, +1) \\
\delta(q_i, 0) &= (q_{i+1}, -1) \text{ if } n + 1 \leq i \leq n + k - 1 \\
\delta(q_i, 1) &= (q_{i+1}, -1) \text{ if } n + 1 \leq i \leq n + k - 1 \\
\delta(q_{n+k}, 1) &= (q_{n+k+1}, +1) \\
\delta(q_{n+k+1}, 0) &= (q_{n+k+1}, +1) \\
\delta(q_{n+k+1}, 1) &= (q_{n+k+1}, +1) \\
\delta(q_{n+k+1}, \$1) &= (f, +1)
\end{aligned}$$

M has $n + k + 3$ states, belongs to $2DFA(k)$ and accepts $L_{n,k}$. \square

Next we want to show that the minimal $1DFA$ for $L_{n,k}$ has at least $c \cdot n \cdot 2^k$ states. To this end, we define a $1DFA$ M , show that M accepts $L_{n,k}$, prove minimality and give an estimate for the number of states of M .

Lemma 4.7. *For every $M \in 1DFA$ with $L(M) = L_{n,k}$, we have $|M| \geq c \cdot n \cdot 2^k$, where $c \approx \frac{1}{2}$.*

Proof. Define $M = (Q, \{0, 1, \$1, \$2\}, \delta, q_0, F)$ by

$$\begin{aligned}
Q &= \{(a_1, \dots, a_k, i) : a_j \in \{0, 1\}, i \in \{0, \dots, n - 1\}, \\
&\quad \#_1(a_1 \dots a_k) \leq i\} \cup \{f\} \\
\delta((a_1, \dots, a_k, i), a) &= \begin{cases} (a_2, \dots, a_k, a, i) & , \quad a = 0 \\ (a_2, \dots, a_k, a, i + 1) & , \quad a = 1, i < n - 1 \\ \text{undefined} & , \quad a = 1, i \geq n - 1 \end{cases} \\
\delta((a_1, \dots, a_k, i), \$1) &= \begin{cases} f & , \quad a_1 = 1 \\ \text{undefined} & \text{else} \end{cases} \\
\delta((a_1, \dots, a_k, i), \$2) &= \begin{cases} f & , \quad i = n - 1 \\ \text{undefined} & \text{else} \end{cases} \\
q_0 &= (0, \dots, 0, 0) \in Q \\
F &= \{f\}.
\end{aligned}$$

Obviously,

$$L(M) = L_{n,k}.$$

In order to prove minimality of M we first show that every state is reachable:

$$\begin{aligned}
q_0 &= \delta(q_0, \varepsilon) \\
f &= \delta(q_0, 1^k \$_1) \\
(a_1, \dots, a_k, i) &= \delta(q_0, 1^{i-l} a_1 \dots a_k), \text{ where } l = \#_1(a_1 \dots a_k)
\end{aligned}$$

Now, we have to show separability of states. (a_1, \dots, a_k, i) and f are separated by $w = \varepsilon$. In order to handle two different (a_1, \dots, a_k, i) and $(\hat{a}_1, \dots, \hat{a}_k, \hat{i})$ we consider two cases.

Case 1. $i \neq \hat{i}$: Wlog, $i > \hat{i}$. Then, the following word separates:

$$\begin{aligned}
\delta((a_1, \dots, a_k, i), 1^{n-i-1} \$_2) &= f, \\
\delta((\hat{a}_1, \dots, \hat{a}_k, \hat{i}), 1^{n-i-1} \$_2) &\neq f,
\end{aligned}$$

since $\$ _2$ ensures that the $\hat{a}_1, \dots, \hat{a}_k$ cannot yield acceptance.

Case 2. $\exists j : a_j = 1, \hat{a}_j = 0$ (wlog)

In this case we have

$$\begin{aligned}
\delta((a_1, \dots, a_k, i), 0^{j-1} \$_1) &= f, \\
\delta((\hat{a}_1, \dots, \hat{a}_k, \hat{i}), 0^{j-1} \$_1) &\neq f.
\end{aligned}$$

Hence, M is minimal and $|M| \approx \frac{1}{2} \cdot n \cdot 2^k$. \square

We summarize our results in the following table. The entry $m_{i,j}$ corresponds to the following trade-off:

$$\begin{array}{c}
i \rightarrow j \\
\leq \\
\geq
\end{array}$$

	2DFA	2DFA(k)	1NFA	1DFA
2DFA		$\leq n^n$ $\geq 2^{n-4} + 1$	$\leq n^n$ $\geq 2^{c \cdot n}$	$\leq n^n$ $\geq 2^{n-3}$
2DFA(k)	$\leq n$ $\geq n$		$\leq n \cdot (k+1) \cdot (\Sigma + 1)^{k+1}$ $\geq n \cdot \frac{1}{c_1 k} \cdot c_2^{\sqrt{k}}$	$\leq n \cdot (k+1) \cdot (\Sigma + 1)^{k+1}$ $\geq c \cdot (n - k - 3) \cdot 2^k$
1NFA	$\leq 2^n$ $\geq ?$	$\leq 2^n$ $\geq 2^{n-2} + 1$		$\leq 2^n$ $\geq 2^n$
1DFA	$\leq n$ $\geq n$	$\leq n$ $\geq n$	$\leq n$ $\geq n$	

REFERENCES

- [1] J. C. Birget : *Basic techniques for two-way finite automata*, Formal Properties of Finite Automata and Applications, Springer-Verlag, LNCS **386** (1989), 56–64
- [2] J. C. Birget : *Concatenation of inputs in a two-way automaton*, Theor. Comput. Sci. **63** (1989), 141–156
- [3] J. C. Birget : *Two-way automaton computations*, RAIRO, Inform. Theor. Appl. **24** (1990), 47–66
- [4] J. C. Birget : *State-complexity of finite state devices, state compressibility and incompressibility*, Math. Syst. Theory **26** (1993), 237–269
- [5] J. C. Birget : *Two-way automata and length-preserving homomorphisms*, Math. Syst. Theory **29** (1996), 191–226
- [6] C. Dwork, L. Stockmeyer : *A time complexity gap for two-way probabilistic finite-state automata*, SIAM J. Comput. **19** (1990), 1011–1023
- [7] J. Goldstine, C. Kintala, D. Wotschke : *On measuring nondeterminism in regular languages*, Inf. Comput. **86** (1990), 179–194
- [8] M. A. Harrison : *Introduction to Formal Language Theory*, Addison-Wesley (1978)

- [9] K. Hashiguchi : *Limitedness theorem on finite automata with distance functions*, J. Comput. System Sci. **24** (1982), 233–244
- [10] J. E. Hopcroft, J. D. Ullmann : *Formal Languages and their Relation to Automata*, Addison-Wesley (1969)
- [11] J. E. Hopcroft, J. D. Ullmann : *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (1979)
- [12] A. R. Meyer, M. J. Fischer : *Economy of description by automata, grammars and formal systems*, 12th Annual SWAT (1971), 188–191
- [13] S. Micali : *Two-way deterministic finite automata are exponentially more succinct than sweeping automata*, Inf. Process. Lett. **12** (1981), 103–105
- [14] M. Rabin, D. Scott : *Finite automata and their decision problems*, IBM J. Research and Development **3** (1959), 114–125
- [15] W. J. Sakoda, M. Sipser : *Nondeterminism and the size of two-way finite automata*, 10th Annual STOC (1978), 275–286
- [16] J. C. Shepherdson : *The reduction of two-way automata to one-way automata*, IBM J. Research and Development **3** (1959), 198–200
- [17] M. Sipser : *Lower bounds on the size of sweeping automata*, 11th Annual STOC (1979), 360–364
- [18] M. Y. Vardi : *A note on the reduction of two-way automata to one-way automata*, Inf. Process. Lett. **30** (1989), 261–264
- [19] D. Wotschke : *Beschreibungskomplexität I*, Lecture Notes, Department of Computer Science, Johann Wolfgang Goethe-Universität, Frankfurt am Main, Germany (1994)

DEPARTMENT OF MATHEMATICS, RICE UNIVERSITY, HOUSTON, TX 77005, USA
E-mail address: `damanik@rice.edu`